Music Generation Conditioned on Emotion

ZHANG Yuyao, Deng Yufan, Zhou Jialu
CSE, DSCT, The Hong Kong University of Science and Technology
yzhangkp, ydengbd, jzhouci@connect.ust.hk

Abstract

Creating new music works is a complex and time-consuming task for human beings. Applying deep neural networks to music creation can greatly improve its effectiveness. However, early work often ignores the role of emotion while generating music. In our project, we will use a Transformer-GAN based model with VAE encoders helping disentangle interpretive features to generate music conditioned on emotions.

1. Introduction

The demand for new musical compositions has increased rapidly as a result of the music industry's tremendous expansion these years. However, creating new music pieces can be a time-consuming and complicated task. To solve the problem, deep neural networks are more and more being applied to music creation in recent years, especially single-track and polyphonic music, as a result of breakthroughs in deep learning techniques. The significance of taking emotional content and particular genres into account while creating music, however, has frequently been disregarded in earlier work in this field. As musical tastes and style needs might fluctuate substantially depending on the situation, this is a severe constraint. Consequently, there is a critical need for a conditional model that can take advantage of the impact of mood and genre on musical composition. These crucial elements can be incorporated into the process of creating music to help us produce more distinctive and varied musical works that satisfy the requirements and expectations of various audiences. The music business will gain from this, and music consumers' general listening pleasure will also be improved.

Previous works provide us with some inspiration, making the generation process interpretable. Followed by a series of works by Music X Lab from NYU Shanghai, EC^2-VAE model [38] makes the variation of one pitch and rhythm while preserving the other possible. [35] decomposed the music into two factors, chord and texture(rhythm, style, etc.) which allows the AI model to "improvise" on a chord progression or "accompany to the singer" as human does. [33] also generate music sequences by finding multiple intrinsic music features from low to high level to make the generation more controllable. When human players are performing, their emotions are reflected in the way they improvise or accompany. These make us think about whether such representations can be leveraged for our goal to generate music with specific emotions.

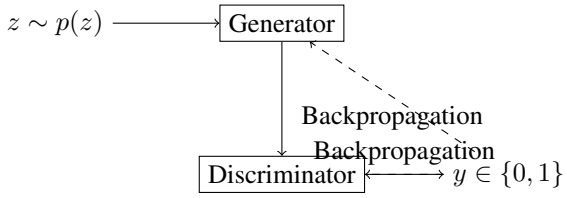


Figure 1. Architecture of a Generative Adversarial Networks (GAN) framework.

Emotion can be quantified along two dimensions — Valence and Arousal [32]. This representation allows the emotion to be added as a condition to LSTM model [34] and Transformer model [26]. In this project, we focus on the task of single-track and polyphonic music generation conditioned with emotion. Inspired by the generation framework conditioning on emotion in [26] and good music feature disentanglement in [35], we want to strengthen the features that are semantically related to emotions to achieve decent emotionally conditioned generation of music. We also examined and evaluated different generation models in this project.

2. Related Work

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GAN) framework is a powerful generative modelling method. A GAN consists of two models, a generator G and a discriminator D. The overall illustration of the GAN framework is shown in Figure 1. Generator G is responsible for generating samples that are indistinguishable from the true samples, while discriminator D is responsible for distinguishing the true sam-

ples from the generated samples. The training process in Gans is adversarial, where the generator G and discriminator D compete against each other. The discriminator D is trained to minimize the binary loss by correctly classifying the input as either true or generated, while the generator G is trained to maximize the binary loss by trying to generate samples that can fool the discriminator into classifying it as true. [11]:

$$\min_G \max_D (\mathbf{E}_x[\log D(x)] + \mathbf{E}_z[\log(1 - D(G(z))])$$

Here x is the real data and z is the input to the generator. This adversarial training process continues until the samples generated by the generator G are generated by the discriminator d that are indistinguishable from the true samples.

2.2. Transformer

The Transformer architecture [36], which was unveiled in 2017, has since grown to be the most well-liked Natural Language Processing (NLP) model and has been effectively used in a variety of different fields, including picture identification and audio processing. A substantial part of its success can be credited to the Self-Attention mechanism, which enables the model to represent relationships between each element in a sequence. Based on a similarity function between the queries and keys, the mechanism computes a weighted sum of the values using matrices of Queries, Keys, and Values. The Linear Transformer [9] is one of the models created to lessen the processing required to use Attention matrices [2, 7, 9, 20]. It makes use of an attention mechanism with linear computational and memory needs that increase linearly as sequence length increases. This is done by the model by replacing the normal softmax similarity score with a more effective factorization of the matrix multiplications required for the attention matrix calculation.

2.3. Music Latent Representation Learning and Disentanglement

Representation learning is an important component of the process of creating new music, as it maps discrete sequences of music and conditions to a continuous latent space. VAEs (Variational Autoencoder) [19] and GANs (Generative Adversarial Networks) [10] are now considered the two best frameworks for music representation learning. Both frameworks are used to construct a latent representation space z between a bidirectional mapping and a sample distribution x such that a new generation of sampled sequences of z. This is done by building a mapping between the distribution of samples x and the latent representation space z. In the last few years, a great deal of work has been carried out within the VAE framework, and VAEs [3, 8, 30, 37] has also achieved significant performance. Several different disentanglement methods have been proposed, [4, 17, 22, 23, 35, 38], in analogy to several musical elements representing the latent space independently, with the aim of providing a better interpretation. For this project, we will use the result of [35] as it makes a separation between chords and textures, both of which carry important emotional information. This will allow us to adopt polyphonic midi sequences.

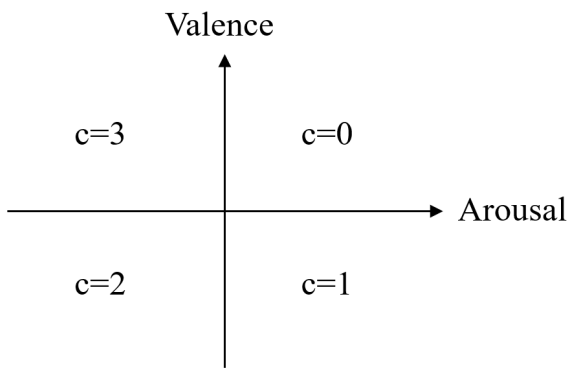


Figure 2. Emotion conditions

3. Data

3.1. Datasets

In this project, following the work by Neves et al. [26] and the work by Wang et al. [38], we intend to employ three datasets, which are AILABS17k [13] and EMOPIA [15].

3.1.1 AILABS17k dataset

The AILABS17k dataset was created by transcribing audio clips of piano performances collected from YouTube using a state-of-the-art piano transcription model [12]. The resulting transcriptions were processed into over 108 hours of MIDI files, comprising the final dataset.

3.1.2 EMOPIA dataset

The EMOPIA dataset is constructed using a similar methodology as AILABS17k, but with a focus on emotional piano performances. The dataset comprises a collection of 387 piano solo performances, in which each MIDI piece has been manually segmented into clips with specific emotion tags.

The emotions are divided into four classes, which correspond to four quadrants of two dimensions coordinate with dimension of valance and arousal. The specific value is shown in Figure 2.

3.2. Data Representation

3.2.1 MIDI-like Events

As proposed in [27], MIDI-like events represent the starting of a note by 'Note ON', and the releasing of if by 'Note OFF'. Similar to the MIDI in music production, it also has another attribute 'Time Shift' which associates each note with time. However, as said in [14], MIDI cannot fully represent information such as bars, beats, and sub-beats. But in real compositions, those features are quite crucial for representing recurrent beat or melody patterns.

3.2.2 REMI

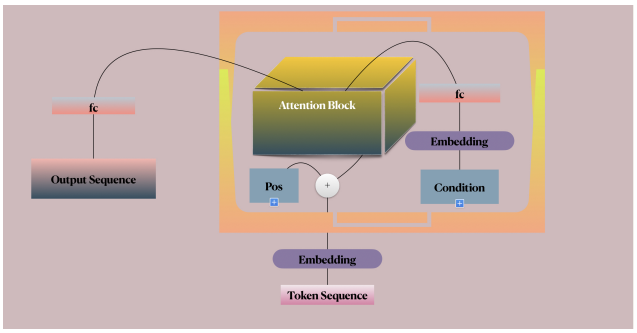
To solve this problem, Huang et al [14] proposed another data representation method called REMI, which stands for REvamped MIDI-derived events. Different from MIDI, the events are represented by token sequences of integers. "None ON", "Note Duration", "Velocity", "Tempo", "Bar", and "Beat" events are mapped to different token values. Using REMI, transformers can generate longer music pieces to minutes, with improved harmony and more coherent structures. For example, event sequence 'Tempo Class(mid), Tempo Value(4), Position(1/16), Note Velocity(11), Note On(57), Note Duration(7)' is represented as '4, 206, 3, 74, 96, 45'.

In this project, we will use REMI to represent the data.

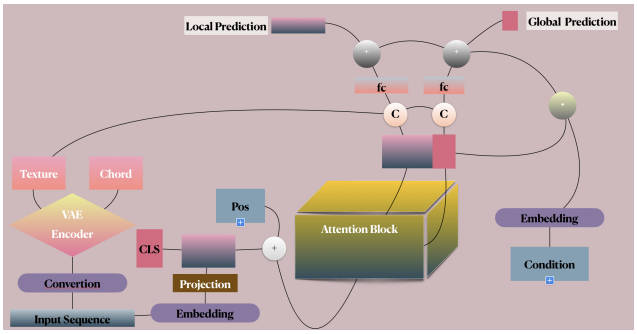
4. Methodology

4.1. Transformer GAN

The overall structure of our model is shown in Figure 3. This structure is based on the model proposed in [26]. The attention blocks in both generator and discriminator are linear versions of the Attention Mechanism [18]. In the generator, the condition of emotion is included by calculating the bias and standard deviation of Layer Normalization [1] based on the input condition. The generator takes in the token sequences and generates the output sequence, which is the music piece. The input sequence to the discriminator is the output sequence of the generator. The discriminator will produce both a global map predicting whether the whole piece is real and satisfy the condition, and a local map predicting whether each local patch in the music piece is real and satisfy the condition [26]. The global prediction corresponds to the CLS token, and the local predictions correspond to 16 local patches equally split from the input sequence. In the discriminator, the condition is included by an inner product between the embedded condition and features, which is according to the projection discriminator in [24]. The discriminator's each global or local output is 1 if the input satisfies the emotion condition and realness simultaneously, and 0 otherwise.



(a) Generator



(b) Discriminator

Figure 3. Overall architecture of our intended model. Generator(above), Discriminator(below). Here 'C' stands for concatenation, '+' for summation, and '·' for inner product. 'CLS' and 'Pos' are used to represent [CLS] token and positional embedding. 'Fc' stands for fully connected layer.

The technique of predicting both globally and locally forces the model to prioritize local structure. Texture, or the rhythm/style, is an important feature that can decide the emotion of the music [28]. We intend to utilize the disentanglement VAE model in [35] to extract the texture features, and let the discriminator to utilize the texture features together with global and local information, in order to improve the performance. For each generated piece, the texture features extracted by the VAE texture encoder is a 256-dimensional feature, which will be concatenated to the output feature map from the attention block before going into the fully-connected classifier. The output feature map from the attention block will have an inner product with the condition and then be added back to the output of the fully-connected layer. The condition in generator will go through a fully-connected layer to get the condition-specific bias and variance for the layer normalization layer. The detailed architecture of the VAE model is described in the next section.

4.2. Pretrained VAE

The overall structure of the trained VAE disentanglement model is shown in Figure 4. We only adopt the texture encoder to extract texture features and do not use the decoders

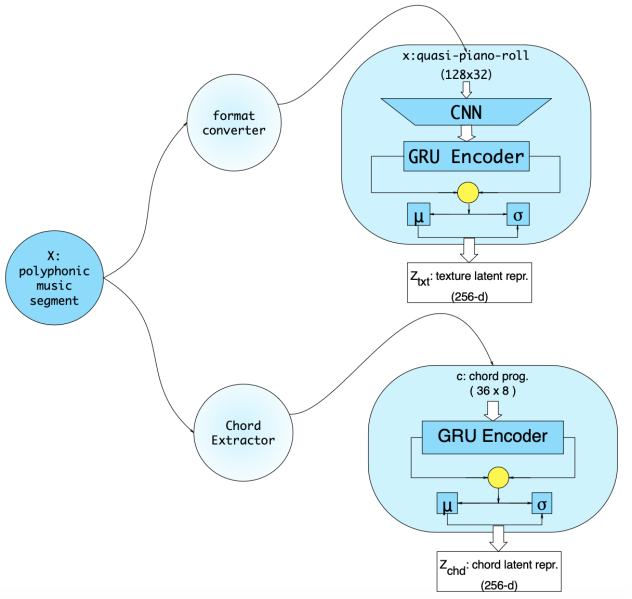


Figure 4. Overall architecture of our VAE framework model. Note that we will use the trained model directed from [35], without further training.

or chord encoder in [35]. Two encoders are described below.

Chord Encoder: The process of the chord encoder begins with extracting the chord progression under one-beat resolution, with the application of rule-based methods [29] [39]. The output of the progression is a matrix that represents the chord information with every column representing a one-beat chord. Each chord itself is a 36-dimensional vector, consisting of three 12-dimensional parts: a one-hot vector indicating the pitch class of the root, a one-hot vector for the bass and a multi-hot chroma vector. Subsequently, the progression is passed through a bi-directional GRU encoder [31] to get the final chord latent representation z_{chd}.

Texture Encoder: The texture encoder encodes the input sequence to a matrix, whose rows of the matrix denote MIDI pitches while columns represent $\frac{1}{4}$ beat. The goal of the texture encoder is to learn a chord-invariant representation of texture by balancing the translation invariance property of convolution and the blurring effect of the max-pooling layer [21]. Then a convolutional module [25] is used in order to extract a blurry "concept sketch" of the texture with minimal underlying chord information. The output goes through a bi-directional GRU encoder to extract the texture representation z_{txt}.

4.3. Training

The training has two stages. In the first stage, the generator will be trained with both EMOPIA and AILABS datasets to gain a good modeling of music. The second stage follows the standard GAN training steps, and the second stage will only use the EMOPIA datasets with emotion labels to let the generator can generate according to the given emotion condition.

For the first stage, the generator will be trained using teacher forcing method similar to language modeling, which is to predict the next word given the input sequence. Under REMI representation, each music event can be seen as a word. The emotion labels is not used in the first stage because AILABS dataset does not have emotional data. This stage aims to provide a good pre-training for generator by utilizing the large quantity of samples without emotion tags in AILABS. In this stage, the input sequences for the generator have lengths of 2048 in REMI representation form. The input is obtained by slicing 2048 length segments from real music pieces in datasets. If the real music is not long enough, padding with 0 and a mask will be added. The REMI representations have value range from 0 to 256, but each value will be represented using one hot representation by a 257-dimensional vector. Therefore, we can use cross-entropy loss for this stage.

$$\mathcal{L}_G = \mathcal{L}_{CE}$$

The first stage will end until convergence.

For the second stage, both the generator and discriminator will be trained in an adversarial style. Also, only the EMOPIA dataset with emotion tags is used and the emotion condition will be input to the generator and discriminator. The generator will firstly generate the fake music of 128-length REMI from 16-length of prime REMI sequence. This prime sequence of 16-length provides only information around 2 notes and 1 tempo, which provides a good initialization for the generator to do auto-regressive generation. The prime sequence is obtained from slicing first 16 elements in the input 128-length sequences. The generated fake sequences will be input to the discriminator and obtain adversarial loss. The adversarial loss we used here is RSGAN loss from [16]. Besides adversarial loss, the teacher forcing training like the first stage with 128-length input will go on at the same time. The cross-entropy loss for teacher forcing training and adversarial loss from discriminator's global and local predictions will be weighted and combined together.

$$\mathcal{L}_G = \mathcal{L}_{CE} + \alpha\mathcal{L}_{RSGAN_G-global} + \beta\mathcal{L}_{RSGAN_G-local}$$

where α and β are hyper parameters and the RSGAN loss for generator is:

$$\mathcal{L}_{RSGAN_G} = -\mathbf{E}_{x,z}[\log(\text{sigmoid}(D(x) - D(G(z))))]$$

The generated fake sequences will be input to the discriminator and the pre-trained VAE texture encoder inside the discriminator.. Before being forwarded through the

VAE texture encoder in the discriminator, the generated REMI sequences will first be converted to MIDI, and then the MIDI will be converted to piano roll matrix, which is the input type of the VAE texture encoder. The gradient back-propagation of the whole VAE model is eliminated. The obtained piano roll will be forwarded through the texture encoder to get the texture feature. The generated REMI sequences will be forwarded through the discriminator, and the discriminator is trained by RSGAN loss plus gradient penalty loss. Both loss will be computed for local and global prediction, and weighted summed.

$$\mathcal{L}_D = \mathcal{L}_{RSGAN_D-global} + \beta\mathcal{L}_{RSGAN_D-local}$$
$$+\gamma(\mathcal{L}_{GP-global} + \mathcal{L}_{GP-local})$$

where β and γ are hyperparameters and the RSGAN loss for the discriminator is:

$$\mathcal{L}_{RSGAN_G} = -\mathbf{E}_{x,z}[\log(\text{sigmoid}(D(G(z)) - D(x)))]$$

and the gradient penalty loss is:

$$\mathcal{L}_{GP} = \mathbf{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D_\phi(\hat{x})\|_2 - 1)^2]$$
$$\hat{x} = \theta G(z) + (1-\theta)x, \theta \in [0,1]$$

in which $\hat{x}$ is a random point sampled between the real data and generated fake data.

5. Experiments

5.1. Evaluation

We will evaluate our approach based on the quality of the generated music clip and the accuracy with which that music clip conveys the target emotional signal. To achieve this, we use both objective and subjective measures for both criteria. For objective evaluation, we used the automatic evaluation indicators proposed in [6] [36]. For subjective assessment, we will use a set of human assessment indicators. We will use these metrics to compare our transformer with the most advanced emotion-constrained symbolic music generation models currently available in the literature.

5.1.1 Objective

We chose Pitch Range (PR), Number of Pitch Classes (NPC), and Polyphony (POLY) for our objective evaluation. Pitch Range is specified as the number of octaves that an instrument or a singer can cover, from the lowest note to the highest. Number of Pitch Classes is the number of unique pitch classes used. Polyphony is the average number of simultaneous notes. We will use the Muspy Library [5] to calculate them. We will generate 400 samples for each model, 100 per class, and evaluate them based on the characteristics we chose above. The average results will then be calculated to determine the overall model score.

5.1.2 Subjective

We use 1) Valence (is the piece negative or positive); 2) Arousal (is it low or high in arousal) to help us evaluate. We will conduct a subjective evaluation of our model by administering a survey to participants, they will be asked to rate the generated music pieces' valence and arousal on a 5-point scale, ranging from very low to very high. Each participant will be assigned 24 musical excerpts, with 8 for each model and 2 for each of the four emotional categories.

5.2. Settings

In experiments, we compared the performance of three models, which are Compound Word Transformer (CWT) in [15], conditional transformers trained in GAN style (TransformerGAN) in [26], and our proposed model with an extra pretrained texture encoder (GANVAE). The CWT is a conditional transformer generator without discriminator, which thus does not have adversarial training. The TransformerGAN is basically our proposed model without the extra pretrained. Our proposed GANVAE is based on the architecture of TransformerGAN [26].

We evaluate both three models both objectively and subjectively according to the evaluation methods mentioned in Section 5. The codes for training and evaluation are all implemented using Pytorch and fast-transformer. The experiments are done with a single GPU of RTX3090. For the training of our proposed GANVAE model, the first pretraining stage is trained for several hours until the loss converges, which is around 100000 steps, and the second adversarial stage is trained for a few days, which is around 25000 steps. The adversarial stage is much slower than the first stage due to the MIDI-Piano roll conversion and discriminator forwarding. The training for TransformerGAN is similar, while CWT without adversarial stage is much faster. The batch size we used is 16, and in the adversarial stage, discriminator is updated once per step. After training, we used the saved model to generate large amount of music with a length of around 30 seconds to 100 seconds. The prime sequences for the generation are valid or real music sequences with length of from 8 to 16. This length of prime sequence can produce a reasonable start point, and will not obey the property of creating new music because the information in prime sequence with 8 to 16 length only contains 1 to 2 notes, which is very subtle compared to the 30 to 100 seconds music that has length around 2000 in REMI sequences form. The evaluation is done on the generated music, and the results are shown in the following sections, which suggests that our

5.3. Objective Evaluation Results

For objective evaluation, we used the automatic evaluation indicators proposed in [6] [36], in details we chose

Method	PR	NPC	Poly
EMOPIA	50.94	8.50	5.60
CWT	49.20	**8.25**	4.36
TransformerGAN	50.43	10.125	**5.08**
GANVAE	**51.23**	9.98	5.04

Table 1. The evaluation results comparing to the original EMOPIA dataset. PR represents Pitch Range, NPC represents the Number of Pitch per Class, and Poly represents Polyphony.

Pitch Range (PR), Number of Pitch Classes (NPC), and Polyphony (POLY). We generated 100 samples for each model, 25 per class, and evaluate them based on the 4 characteristics. The results are shown in Table 1.

It can be seen from the table, our 'GAN-VAE', has slightly better performance for pitch range but is worse in the number of pitch classes and polyphony(the closer to the statistics of the training set EMOPIA the better). However, our model contains another module, so the number of parameters is greater than the Transformer GAN model while the performance is similar, which suggests that the proposed component is not very effective. CWT has better number of pitch classes, but has much lower pitch range and polyphony.

5.4. Subjective Evaluation Results

In order to subjectively evaluate our model, we created a survey. Participants were asked to assess the musical excerpts generated by the model based on their overall quality and their ability to evoke the desired emotion. They assessed the characteristics of the samples using a 5-point Likert scale ranging from very low to high. Each participant had to listen to 24 pieces of music, eight for each model and two for each of the four divisions. The average scores in Figure 5 for each model's portraits, originality, structure, and overall quality indicate that our proposed GANVAE has similar competitiveness in high arousal and low valence to the other methods, but TransGAN outperforms ours in the other emotion conditions, according to our survey.

Figure 5 depicts the results of comparing the participants' responses to valence and arousal queries with the actual emotion labels used in the generation process. Our findings indicate that models have a more difficult time encoding valence than arousal, implying that certain musical aspects are easier for neural networks to comprehend than others.

Moreover, we see that, in general, our model is similar to the baseline. By looking at the boxplots, between the three, it seems that CWT and TransformerGAN sometimes produce samples that place themselves more strongly on their theoretically relevant side, and GANVAE performs similarly to these two models, placing more than half of excerpts on the correct side of the midline.

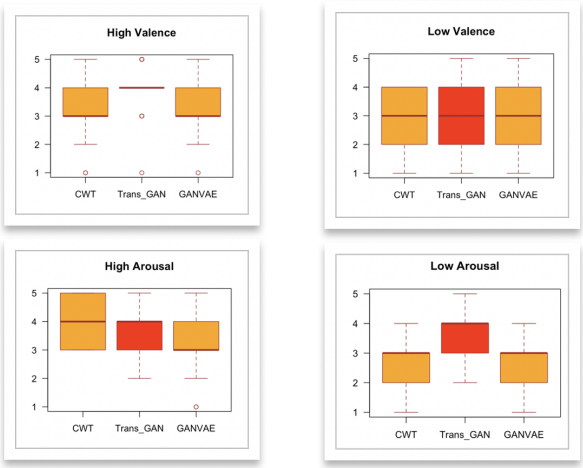


Figure 5. Results of the experiment where participants rated musical samples according to their perceptions about valence and arousal. The CWT, TransGAN, and GANVAE correspond, respectively, to the Baseline model, Transformer GAN and Transformer-GAN with VAE.

5.5. Discussion

The performance of the model with VAE texture encoder added doesn't have a decent difference from the one without VAE texture encoder. We manually checked the music generated by the generator during the training and analysis it to find a possible explanation. Since the length of generated REMI during the GAN stage is 128, which usually contain around 32 notes after being converted into MIDI. The piece is around 4 to 5 seconds. We think that MIDI in this length is either inconsistent with the midi file used to train the pre-trained VAE encoder, or is too short to get some useful information for the discriminator. Another reason is that the higher level information contained in the texture embedding space may not be consistent with the representation the Transformer GAN can learn, in the future it is meaningful to further investigate the higher level representation of the music.

In addition, we find that many of our generated pieces are like chord bars. However, there are various patterns in real performance. We chose one example and put it in Figure 6. In the future, researchers may also lay more emphasis on generating music with more patterns and styles, even making them controllable.

The supplementary zip file contains generated samples in MIDI format. We also include a few samples in MP3 format. However, the MP3 format is too large, so we only include a few.

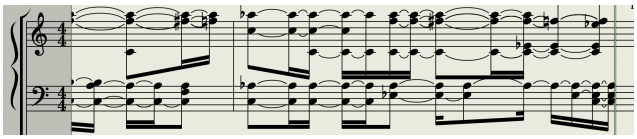


Figure 6. Results of generated pieces' sheet. one can see that the generated music is only chord bars.

6. Conclusion

We tried to improve the performance of the Transformer-GAN model to generate music pieces with certain condition labels. Via disentangling and enhancing the texture information through a pre-trained VAE model, we achieved similar performance with the state-of-the-art Transformer GAN model. Although there is no improvement, we believe that this idea can work by either finding a better way to match the embeddings of the texture features and the generated sequences or just fine-tuning parameters. Our project also provides an idea of fusing more emotional information through the disentanglement of the musical characteristics(i.e. texture, chord, beat) when conditioning on emotion labels, which may be a potential direction for future research.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[2] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer, 2018. 2

[3] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer. *arXiv preprint arXiv:1809.07600*, 2018. 2

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016. 2

[5] Hao-Wen Dong, Ke Chen, Julian McAuley, and Taylor Berg-Kirkpatrick. Muspy: A toolkit for symbolic music generation, 08 2020. 5

[6] Hao-Wen Dong and yi-hsuan Yang. Convolutional generative adversarial networks with binary neurons for polyphonic music generation. 09 2018. 5

[7] Philippe Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces. In *International Society for Music Information Retrieval Conference*, 2018. 2

[8] Philippe Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces. In *ISMIR*, pages 175–181, 2018. 2

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. 2

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2

[12] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017. 2

[13] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 178–186, 2021. 2

[14] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020. 3

[15] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. *arXiv preprint arXiv:2108.01374*, 2021. 2, 5

[16] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 4

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 3

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 2

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 4

[22] Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018. 2

[23] Yin-Jyun Luo, Kat Agres, and Dorien Herremans. Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders. *arXiv preprint arXiv:1906.08152*, 2019. 2

[24] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018. 3

[25] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress. 4

[26] Pedro Neves, Jose Fornari, and João Florindo. Generating music with sentiment using transformer-gans. *arXiv preprint arXiv:2212.11134*, 2022. 1, 2, 3, 5

[27] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32:955–967, 2020. 3

[28] Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, 2020. 3

[29] Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Comput. Music J.*, 26(2):27–49, jul 2002. 4

[30] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning*, pages 4364–4373. PMLR, 2018. 2

[31] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music, 2019. 4

[32] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980. 1

[33] Hao Hao Tan and Dorien Herremans. Music fadernets: Controllable music generation based on high-level features via low-level feature modelling. *arXiv preprint arXiv:2007.15474*, 2020. 1

[34] Xiaodong Tan, Mathis Antony, and H Kong. Automated music generation for visual art through emotion. In *ICCC*, pages 247–250, 2020. 1

[35] Ziyu Wang, Dingsu Wang, Yixiao Zhang, and Gus Xia. Learning interpretable representation for controllable polyphonic music generation. *arXiv preprint arXiv:2008.07122*, 2020. 1, 2, 3, 4

[36] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Comput. Appl.*, 32(9):4773–4784, may 2020. 2, 5

[37] Ruihan Yang, Tianyao Chen, Yiyi Zhang, and Gus Xia. Inspecting and interacting with meaningful music representations using vae. *arXiv preprint arXiv:1904.08842*, 2019. 2

[38] Ruihan Yang, Dingsu Wang, Ziyu Wang, Tianyao Chen, Junyan Jiang, and Gus Xia. Deep music analogy via latent representation disentanglement. *arXiv preprint arXiv:1906.03626*, 2019. 1, 2

[39] Adrien Ycart, Lele Liu, Emmanouil Benetos, and Marcus Pearce. Investigating the perceptual validity of evaluation metrics for automatic piano music transcription. *Transactions of the International Society for Music Information Retrieval*, 3:68–81, 06 2020. 4